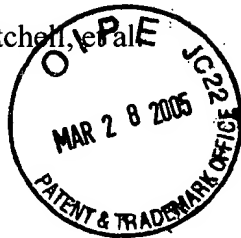


IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

AFI 2172 ^{72W}

In re application of: Todd A. Mitchell, et al. : Date: March 22, 2005
Group Art Unit: 2172 : IBM Corporation
Examiner: T. Nguyen : Intellectual Property Law
Serial No.: 09/488,738 : Dept. 917, Bldg. 006-1
Filed: January 20, 2000 : 3605 Highway 52 North
Title: USER INTERFACE FOR AUTOMATED : Rochester, MN 55901
PROJECT MANAGEMENT



To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF IN SUPPORT OF APPEAL
FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

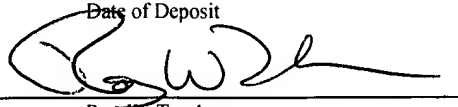
Applicant herewith submits an appeal brief in support of the appeal to the Board of Appeals from the decision dated October 22, 2004, of the Primary Examiner finally rejecting claims 1-18.

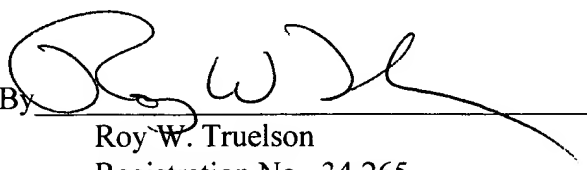
The appeal brief fee of \$500.00 is to be charged to Deposit Account No. **09-0465**. A duplicate copy of this sheet is enclosed.

Date: March 22, 2005

Respectfully submitted,

TODD A. MITCHELL, et al.

<p>CERTIFICATE OF MAILING UNDER 37 CFR 1.8(a)</p> <p>I hereby certify that the enclosed or attached correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on</p> <p>March 22, 2005 Date of Deposit</p> <p> Roy W. Truelson</p>
--

By 
Roy W. Truelson
Registration No. 34,265

Voice/Fax: (507) 289-6256

03/28/2005 GWORDDF1 00000053 090465 09488738

01 FC:1402 500.00 DA

Docket No. RO999-164
Serial No. 09/488,738



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Todd A. Mitchell, et al. : Date: March 22, 2005
Group Art Unit: 2172 : IBM Corporation
Examiner: T. Nguyen : Intellectual Property Law
Serial No.: 09/488,738 : Dept. 917, Bldg. 006-1
Filed: January 20, 2000 : 3605 Highway 52 North
Title: USER INTERFACE FOR AUTOMATED : Rochester, MN 55901
PROJECT MANAGEMENT

Commissioner for Patents
P.O. Bos 1450
Alexandria, VA 223313-1450

**APPEAL BRIEF IN SUPPORT OF APPEAL
FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS**

Sir:

This is an appeal of a Final Rejection under 35 U.S.C. §103(a) of claims 1-18 of Application Serial No. 09/488,738, filed January 20, 2000. This brief is submitted pursuant to a Notice of Appeal filed January 24, 2005, as required by 37 C.F.R. §1.192.

1. Real Party in Interest

International Business Machines Corporation of Armonk, NY, is the real party in interest. The inventors assigned their interest as recorded on January 20, 2000, on Reel 010562, Frame 0242.

Docket No. RO999-164
Serial No. 09/488,738

2. Related Appeals and Interferences

There are no related appeals nor interferences pending with this application.¹

3. Status of Claims

Claims 1-18 are pending and stand finally rejected. The claims on appeal are set forth in Appendix A.

4. Status of Amendments

No amendments were filed following Final Rejection on October 22, 2004.

5. Summary of Invention

The invention herein relates to a user interface for interactive project management software of a type which supports the performance of multiple tasks in furtherance of projects responsive to user selections. In the case of large, complex projects, it is known to have multiple groups or sub-groups of users assigned to the performance of respective sets or subsets of tasks using the project management software (Spec p. 2, lines 17 - p. 3, line 7). In accordance with appellants' invention, each such group is provided its own unique

¹ A final rejection in the present application dated August 20, 2002, was previously appealed. Following appellants' submission of their appeal brief dated January 10, 2003, the Examiner withdrew the finality of the rejection and re-opened prosecution. The earlier appeal is therefore no longer pending. The issues presented in the present appeal are similar to those of the earlier appeal, although there has been some intervening amendment of the claims and the Examiner relies on one newly cited reference.

interactive interface to the project management software (Spec. p. 4, lines 2-5). Specifically, a process management function supports the interactive definition of multiple user groups, and further supports the interactive definition of a respective different group interface to the project management software for each group (Spec. p. 4, lines 2-15; p. 11, lines 16-25; p. 16; lines 21- p. 17, line 23; Figs. 12 & 13). Thus, the set of selectable task functions presented to users in a first group is different from (although possibly overlapping) the set of selectable task functions presented to users in a second group, even though both groups use the same project management software (Spec. p. 14, line 14 - p. 15, line 1; Figs. 9-11). It is further possible to assign different labels to the same function in the different interactive interfaces of different groups (Spec. p. 4, lines 14-18; p. 15, lines 2-4).

6. Issues

Claims 1-18 are finally rejected under 35 U.S.C. §103(a) as unpatentable over Knudson et al. (U.S. Patent 5,765,140), in view of Nakaoka (U.S. Patent 6,092,048). The only issue in this appeal is whether the claims are prima facie obvious in view of *Knudson* and *Nakaoka*.

7. Grouping of Claims

Appellants expressly state that, for purposes of appealing the grounds of rejection advanced by the Examiner herein, all claims stand or fall together. However, in the event that new references are cited or new arguments advanced for rejection of the claims, appellants reserve the right to argue that claims do not stand or fall together.

8. Argument

Appellants assert that the Examiner failed to establish adequate grounds of rejection for the following reason:

The mere aggregation of *Knudson* and *Nakaoka* does not meet the limitations of appellants' claims without further modification to the proposed combination, and a suggestion to make such a further modification is lacking.

Overview of Invention

A brief overview of appellants' invention in light of existing art will be helpful in appreciating the issues herein. Appellants' invention is in the realm of user interface, and provides a method of customizing a user interface for complex project management software, so that the user interface is optimized for each group of users.

In the realm of user interface, it is often the case that a useful, new and unobvious invention does not provide the user with any new capability to perform some action which could not previously be performed by other means, but instead, provides the user with the capability to perform the action in a manner which is more efficient, more natural, easier to learn, easier to implement and/or in some other respect, better, from the user interface perspective, than prior art techniques. This distinction is a subtle but important one. It may be observed, for example, that the ubiquity of so-called "personal computers" is due in large part to the fact that graphical user interfaces have made use of such systems comfortable to the average person, who lacks skilled training as a typist or computer operator. However, in general such GUI's do not provide the user with any new capability which did not previously exist. Almost all system tasks invoked using a GUI interface can also be invoked using older text-based interfaces.

Appellants' invention addresses a specific area of user interface, namely, the user interface for project management software which is shared by multiple users. Typically, such software accesses a common database and enables its users to perform some set of functions ("tasks") using the database to complete one or more projects. Where large, complex projects are involved, the number of different functions performed by the users can be quite large. It is common in such cases to divide responsibilities for performing the different functions among different users or groups of users. Each group of one or more users is therefore responsible for performing some subset of the full set of functions available for completing the project, and the subset of functions used by each group is different from that of any other group, although the subsets of different groups may overlap. In a typical such case, there is considerable commonality among the various groups, such as a need to access the common database and perform certain common functions in relation thereto. Other functions may be particular to a single group of users or to some subset of all the groups. Appellants have further observed that different groups performing tasks using a common database sometimes have a different terminology for the same task.

Conventionally, there are typically two approaches to handling of large projects involving different groups with different responsibilities. It is possible to create a single project management application used by all groups, or it is possible to create separate applications used by each respective group.

Conventionally, where a single application is used by different groups with different responsibilities, the project management software provides a standard interface, which may be a single interactive menu or a set of interactive menus. This standard interface may be considered a logical "OR" of the user interface requirements of each separate group of users. I.e., any function required by at least one of the user groups is provided by the

standard interface. This means that the users of any given group are presented with all available interactive user function selections in the standard interface. Frequently, many of these selections are not even used by the particular group. While a single common interface does achieve certain goals of uniformity, and can reduce the development costs vis-a-vis developing and maintaining separate interfaces, there are certain drawbacks to this approach. Specifically, the presentation of a large number of function selections unrelated to the responsibilities of a particular group makes the interface overly complex and difficult to use.

It is, of course, possible to write separate applications for each group, or to write custom computer programming code providing different interfaces within a common application for different groups of users, so that each group has its own unique interface. However, the cost of maintaining different applications having different interfaces, or of creating custom programming code to provide a separate interface for each group from a common application, discourages this approach.

In accordance with appellants' invention, the project management software includes a custom user interface definition feature, whereby custom user interfaces for different groups of users may be interactively defined. A custom interface presents a user with only the interactive function selections which are applicable to the user's group, and hence useful to the user; superfluous functions used only by other groups are not presented. The user thus is provided a simplified, easier to understand interactive interface. The capability to flexibly and easily define different group user interfaces is supported by using interface definitions. These are editable data structures (not directly executable code) which define customization parameters for use by the interface generator (which is an executable

program), so that different interfaces are presented to users from different groups.² I.e., the interface definitions are directives to the interface generator; while not directly executable code, they cause the interface generator to do something.

The capability to generate custom interactive interfaces for different user groups from group-specific interface definitions is a significant feature of appellants' invention.

The mere aggregation of *Knudson* and *Nakaoka* does not meet the limitations of appellants' claims without further modification to the proposed combination, and a suggestion to make such a further modification is lacking.

In order to support a rejection for obviousness, there must be some suggestion in the art to combine the references in such a manner as to form each and every element of appellants' claimed invention. It is not sufficient that a suggestion may exist to combine the references, if such a combination does not meet the limitations of appellants' claims without some further non-obvious modification. Both *Knudson* and *Nakaoka* deal with similar subject matter, and appellants do not challenge the combination of the two references per se. But such a hypothetical combination neither meets the critical limitations of appellants' claims nor suggests the modifications necessary to construct these critical elements.

² In the preferred embodiment, the custom interface for the user's group is simply one interface available to the user, and the user still has the capability to access a "standard interface" containing all functions, in the unlikely event that the user needs to access a function normally used by other groups. Thus, appellants' invention is not intended as a security device to prevent access to certain functions, but is an improvement to the interface intended to make the user more productive by showing the user only those function selections which the user is most likely to want.

Knudson discloses a “project management system”, in which tasks to be performed, schedules, funding and similar information is tracked in a database. The exemplary information being tracked in *Knudson* is accounting data, such as time sheets which record the number of person hours expended in connection with a task. *Knudson* discloses that different personnel associated with a project might have different schedules and different tasks to perform, and that this information might be tracked in the database. *Knudson*’s database information can be displayed to users, thus showing who is responsible for performing certain tasks and so forth.

Nakaoka similarly discloses a “task execution support system”, which tracks information concerning tasks performed and to be performed by different personnel in an organization. I.e., *Nakaoka* teaches a database containing information such as a task identifier, task status, task description, and responsible person for multiple tasks performed within the organization. A larger task may be further subdivided into a hierarchy of sub-tasks, each subtask having its own information in the database. The thrust of *Nakaoka*’s invention appears to be its ability to handle certain indefinite tasks, i.e. tasks in which not all of the elements are defined at the time the task is initiated, and for which the definition of certain elements depends on the outcomes of earlier tasks. *Nakaoka* lists examples of tasks, such as “go on a business trip”, “write report”, “clear travel expenses”, etc. The “tasks” are thus not functions performed directly by a computer system, although, depending on the task, a computer system might be used to assist in completing the task. For example, a report might be written using conventional word processing software on a computer system, hotel and airplane reservations for a business trip might be made using computers, etc.

Nakaoka discloses that different persons perform different tasks. As disclosed in *Nakaoka*, a database user can select a particular person as a key variable, and the system

will display all tasks assigned to that person. This is a well-known feature of conventional databases.

Taken together, what do these references show? Both references show essentially that a database may be used to keep track of project information, that project information may include assignments of particular individuals to perform particular tasks, and that in a large project different people will perform different tasks.

Does such a combination include all the elements of appellants' claims? Manifestly, it does not, for the very essence of appellants' invention, the most crucial part of the claims, is the lacking from such a combination. I.e., the proposed combination simply provides standard complex user interface for maintaining a lot of data about a project, wherein the data may include the fact that different people perform different tasks. But there is no disclosure or suggestion in either reference of using *interface definitions* (i.e., editable data objects) as directives to an interface generator for generating *different user interfaces for different groups of users*.

The Examiner's rejection, as near as appellants can understand it, equates the display of data with an "interface". What the references show is the display of data from a database. The data itself reflects the reality of large project management, specifically, that different groups of people perform different tasks. Therefore, it is possible to display data from the database showing different groups of people, different tasks to which they are assigned, the status of the tasks, and so forth. The Examiner appears to consider these displays "interfaces", and finds various elements from the references based on this line of reasoning.

It would be useless to quibble about the meaning of the word “interface” in the abstract. Standing by itself, the word could mean many things, and the display of data falls within the general scope of an “interface” in at least some contexts. But claims are never devoid of context, and as used in the claims herein, the “interface” is something completely different.

Appellants are not asking this Board to import limitations from the specification into the claims. Appellants only ask that clear and explicit limitations on the word “interface”, contained in the claims themselves, be given effect. Appellants’ representative claim 1 recites in part:

1. A method for managing a project requiring a plurality of *tasks performed on at least one computer system* by a plurality of users, said at least one computer system containing a *process interface supporting a pre-defined set of task actions* performed by said at least one computer system with respect to data objects stored on said at least one computer system, the method comprising the steps of:
 - interactively defining a plurality of groups of users ... ;
 - interactively generating, for each of said plurality of groups of users, a respective project tracking interface definition, *each project tracking interface definition being a data object defining a respective set of task selections, each task selection of a set of task selections corresponding to a respective task action of said pre-defined set of task actions performed by said at least one computer system with respect to a respective one or more said data objects stored on said at least one computer system*, wherein a first set of task selections of a first project tracking interface for a first group of users is different from a second set of task selections of a second project tracking interface for a second group of users;
 - associating a first user with said first group of users;
 - presenting said first project tracking interface ... to said first user;
 - performing task actions ... responsive to said first user interactively selecting the corresponding task selections of said first set of task selections;
 - associating a second user with said second group of users;
 - presenting said second project tracking interface ... to said second user; and
 - performing task actions ... responsive to said second user interactively selecting the corresponding task selections of said second set of task selections.
- [emphasis added]

The remaining independent claims vary in their language, but all recite the essential feature of interactively defining separate and different project tracking interfaces for separate user groups, the interface being an interactive interface from which a user makes selections to perform “task actions” by a computer system.

As used above, a “process interface” supports a “pre-defined set of task actions performed by the computer system”. The “project tracking interface definitions” define a set of “task selections” corresponding to a subset of these “pre-defined task actions”. “Interface” as used by appellants’ claims therefore refers to something which invokes functions on the computer.

Nakaoka does indeed use the word “interface” in connection with the display of data, but a careful reading of the relevant passage from *Nakaoka* discloses that its “interface” is nothing more than a display of data with the capability to interactively perform operations on the displayed data, such as editing the data.³ Operations on data in a

³ The following passage from *Nakaoka*, relied upon by the Examiner, explains the “interface”:

“... The service provider machine 25010 provides information concerning task information, and processes the operation of the task information. Each of the client machines 25070 includes the task display/operation unit 1030, displays the task information to the user, and provides a user interface which enables the user to operate the task information. A network 25050 enables the client machine 25070 to exchange information therebetween.

“An information display unit 1510 in the task information display/operation unit 1030 provides a user interface which enables task information to be displayed and operated, and includes the following elements: The information display unit 1510 includes a task list display section 1520 for displaying registered tasks in the form of a list and which provides a user interface of an operation concerning a task entry, a task property display section 1530 for providing a user interface of the operation concerning a task property and a task action display section 1540 for displaying a task action of a selected task and which provides a user interface of the operation concerning a task

database are indeed computer functions, but if one attempts to read this on appellants' claimed invention, the only "pre-defined set of task actions" disclosed would be the database operations themselves. ***But these operations do not change for different groups of users.*** I.e., to the extent that the database interface of *Nakaoka* is considered a "process interface" within the meaning of appellants' claims, there is no disclosure or suggestion in *Nakaoka* of "project tracking interface definitions" which define subsets of task selections, i.e. subsets of the database functions, which are associated with different groups of users of the database. *Nakaoka* only discloses that a single database interface exists, and nothing more. This database interface is not changed just because different data can be displayed and manipulated.

Knudson is similar in import to *Nakaoka*. It discloses a database which tracks project related information, including assignment of different people to different tasks and so forth. But like *Nakaoka*, *Knudson* does not disclose multiple *interfaces* applicable to different respective groups of people performing tasks. The Examiner appears to take the position that any time different data appears on a display screen, a different "interface" is presented. As explained previously with respect to *Nakaoka*, the Examiner is not entitled to ignore explicit limitations on the word "interface" contained in the claims. As used in the claims, a "process interface" supports "task actions performed by the computer", and the interface definition defines subsets of the task actions which are different for different groups of users.

property. Also, the task information display/operation unit 1030 includes a keyboard 1550 for enabling the user to enter data and a mouse 1560 for enabling the user to move or operate a cursor." [Col. 8, lines 41-65]

Both *Knudson* and *Nakaoka* disclose the *display* of data concerning tasks, and this data is different for different groups of users. But these “tasks” are not computer functions at all; they are tasks which are being tracked by a database in a computer system. Appellants’ claims require something quite different, i.e., “task actions performed by a computer”. A “project tracking interface definition” defines a set of “task selections” in an interface, i.e. something the user *selects* to invoke a *task action performed by the computer*.

Neither cited reference discloses that the user interface is customizable. Neither reference discloses customization of the user interface by interactively editing an interface definition file, which is used to generate the interface. Neither reference discloses that different user groups would benefit from different user interfaces. Neither reference discloses that different interfaces of any type are presented to different groups of users. On the contrary, in the case of both references, the same, fixed database interface is presented to each and every user, whoever he may be, and the only thing that changes is the data itself which is displayed using the interface.

It is too easy to find the elements of an invention in hindsight. The Examiner has rather arbitrarily mixed features of the references to find make a case of obviousness. On the one hand, a database contains a user interface with selectable functions which can be performed on the data, hence elements of a “process interface” are present. On the other hand, the data contained in the database reveals that different tasks are performed by different users. What do these elements in combination really suggest? It is a bit far-fetched to say that, because the database data shows that different personnel perform different tasks, it would be obvious to create separate interface definition files for different

groups of users, each interface definition file defining its own unique subset of selectable functions to be presented to a user in a custom interface.

The fact that different persons or groups of persons involved in a complex project may perform different tasks is well known. The fact that you can track almost any data in a database is known. A variety of interactive interfaces, from the simple to the complex, are known. But the motivation or suggestion to provide a capability to customize the project management interface, so that different groups associated with a project are presented with different interfaces, is not shown in either reference. The suggestion to make this particular modification has come from only one place: appellants' disclosure.

9. Summary

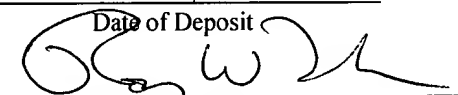
Appellants disclose project management software in which it is possible to interactively customize the user interface presented to different groups of users associated with the project, and to thereafter present the interface optimized for a particular group to users in that group, thereby improving user productivity. The key fact that different user interfaces, i.e. different sets of selectable functions performed by the computer, are interactively defined in an interface definition data object, and presented to different groups of users, is neither taught nor suggested by either cited reference. A combination of different elements in the references can not show this recited capability without some further modification not suggested in the art. The only suggestion to make the further modification, to provide different user interfaces for different groups of users, comes from appellants' disclosure.

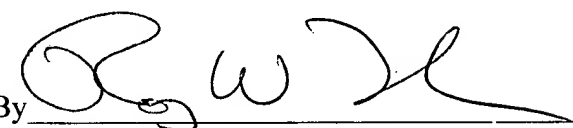
For all the reasons stated herein, the rejection for obviousness was improper, and appellants respectfully request that the Examiner's rejection of the claims be reversed.

Date: March 22, 2005

Respectfully submitted,

TODD A. MITCHELL, et al.

<p style="text-align: center;">CERTIFICATE OF MAILING UNDER 37 CFR 1.8(a)</p> <p>I hereby certify that the enclosed or attached correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on</p> <p style="text-align: center;">March 22, 2005 Date of Deposit</p> <p style="text-align: center;"> Roy W. Truelson</p>
--

By 
Roy W. Truelson, Attorney
Registration No. 34,265

From:

IBM Corporation
Intellectual Property Law
Dept. 917, Bldg. 006-1
3605 Highway 52 North
Rochester, MN 55901

Voice/Fax: (507) 289-6256

APPENDIX A (CLAIMS)

1 1. A method for managing a project requiring a plurality of tasks performed on at least
2 one computer system by a plurality of users, said at least one computer system containing a
3 process interface supporting a pre-defined set of task actions performed by said at least one
4 computer system with respect to data objects stored on said at least one computer system,
5 the method comprising the steps of:

6 interactively defining a plurality of groups of users associated with the project;

7 interactively generating, for each of said plurality of groups of users, a respective
8 project tracking interface definition, each project tracking interface definition being a data
9 object defining a respective set of task selections, each task selection of a set of task
10 selections corresponding to a respective task action of said pre-defined set of task actions
11 performed by said at least one computer system with respect to a respective one or more
12 said data objects stored on said at least one computer system, wherein a first set of task
13 selections of a first project tracking interface for a first group of users is different from a
14 second set of task selections of a second project tracking interface for a second group of
15 users;

16 associating a first user with said first group of users;

17 presenting said first project tracking interface having said first set of task selections
18 to said first user;

19 performing task actions corresponding to task selections of said first set of task
20 selections responsive to said first user interactively selecting the corresponding task
21 selections of said first set of task selections;

22 associating a second user with said second group of users;

23 presenting said second project tracking interface having said second set of task
24 selections to said second user; and

25 performing task actions corresponding to task selections of said second set of task
26 selections responsive to said second user interactively selecting the corresponding task
27 selections of said second set of task selections.

1 2. The method of claim 1, wherein each said project tracking interface definition
2 comprises a respective task description, whereby a task selection for a first project tracking
3 interface may have a first task description, and the same task selection for a second project
4 tracking interface may have a second task description different from said first task
5 description.

1 3. The method of claim 1, wherein each task selection displayed in a project tracking
2 interface includes a task status indicator.

1 4. The method of claim 3, wherein said task status indicator is assumes one of a
2 plurality of colors, each color corresponding to a respective status.

1 5. The method of claim 1, wherein each said project tracking interface definition is an
2 interface definition file containing entries corresponding to tasks, wherein a first interface
3 definition file for defining said first project tracking interface contains a respective entry for
4 each task selection in said first set of task selections, and a second interface definition file
5 for defining said second project tracking interface contains a respective entry for each task
6 selection in said second set of task selections.

1 6. The method of claim 5, wherein each said entry in an interface definition file
2 includes a respective task description field, whereby a task selection for said first project
3 tracking interface may have a first task description, and the same task selection for said
4 second project tracking interface may have a second task description different from said
5 first task description.

1 7. The method of claim 5, wherein each said entry in an interface definition file
2 includes a respective scope field specifying the scope of the task selection, whereby a task
3 selection for said first project tracking interface may have a first scope, and the same task
4 selection for said second project tracking interface may have a second scope different from
5 said first scope.

1 8. A computer program product for managing a project requiring a plurality of tasks
2 performed on at least one computer system by a plurality of users, said at least one
3 computer system containing a process interface supporting a pre-defined set of task actions
4 performed by said at least one computer system with respect to data objects stored on said at
5 least one computer system, said computer program product comprising:

6 a plurality of processor executable instructions recorded on signal-bearing media,
7 wherein said instructions, when executed by at least one processor, cause at least one
8 computer to perform the steps of:

9 receiving interactive input defining a plurality of groups of users associated with the
10 project;

11 receiving interactive input generating, for each of said plurality of groups of users, a
12 respective project tracking interface definition, each project tracking interface definition
13 being a data object defining a respective set of task selections, each task selection of a set of
14 task selections corresponding to a respective task action of said pre-defined set of task

15 actions performed by said at least one computer system with respect to one or more said
16 data objects stored on said at least one computer system, wherein a first set of task
17 selections of a first project tracking interface for a first group of users is different from a
18 second set of task selections of a second project tracking interface for a second group of
19 users;

20 associating a first user with said first group of users;
21 presenting said first project tracking interface having said first set of task selections
22 to said first user;

23 invoking task actions corresponding to task selections of said first set of task
24 selections responsive to receiving interactive input from said first user selecting the
25 corresponding task selections of said first set of task selections;

26 associating a second user with said second group of users;
27 presenting said second project tracking interface having said second set of task
28 selections to said second user; and

29 invoking task actions corresponding to task selections of said second set of task
30 selections responsive to receiving interactive input from said second user selecting the
31 corresponding task selections of said second set of task selections.

1 9. The program product of claim 8, wherein each said project tracking interface
2 definition comprises a respective task description, whereby a task selection for a first project
3 tracking interface may have a first task description, and the same task selection for a second
4 project tracking interface may have a second task description different from said first task
5 description.

1 10. The program product of claim 8, wherein each task selection displayed in a project
2 tracking interface includes a task status indicator.

1 11. The program product of claim 10, wherein said task status indicator is assumes one
2 of a plurality of colors, each color corresponding to a respective status.

1 12. The program product of claim 8, wherein each said project tracking interface
2 definition is an interface definition file containing entries corresponding to tasks, wherein a
3 first interface definition file for defining said first project tracking interface contains a
4 respective entry for each task selection in said first set of task selections, and a second
5 interface definition file for defining said second project tracking interface contains a
6 respective entry for each task selection in said second set of task selections.

1 13. The program product of claim 12, wherein each said entry in an interface definition
2 file includes a respective task description field, whereby a task selection for said first project
3 tracking interface may have a first task description, and the same task selection for said
4 second project tracking interface may have a second task description different from said
5 first task description.

6 14. The program product of claim 13, wherein each said entry in an interface definition
7 file includes a respective scope field specifying the scope of the task selection, whereby a
8 task selection for said first project tracking interface may have a first scope, and the same
9 task selection for said second project tracking interface may have a second scope different
10 from said first scope.

1 15. A computer program product for managing a project requiring a plurality of tasks
2 performed on at least one computer system by a plurality of users, said at least one
3 computer system containing a process interface supporting a pre-defined set of task actions
4 performed by said at least one computer system with respect to data objects stored on said at
5 least one computer system, said computer program product comprising a plurality of
6 processor executable instructions recorded on signal-bearing media, said instructions
7 comprising:

8 an interface definition access function, said interface definition access function
9 accessing a project tracking interface definition, said project tracking interface definition
10 being one of a plurality of project tracking interface definitions, each said project tracking
11 interface definition being associated with a respective group of users of said plurality of
12 users, each project tracking interface definition being a data object defining a respective set
13 of task selections, each task selection of a set of task selections corresponding to a
14 respective task action of said pre-defined set of task actions, wherein a first set of task
15 selections of said first project tracking interface definition for a first group of users is
16 different from a second set of task selections of a second project tracking interface
17 definition for a second group of users; and

18 a project tracking interface generator, said generator generating a project tracking
19 interface defined by a project tracking interface definition of said plurality of project
20 tracking interface definitions, said project tracking interface defined by a project tracking
21 interface definition presenting a user with the set of task selections of the project interface
22 definition and allowing the user to invoke task actions corresponding to respective task
23 selections presented to the user by interactively selecting the corresponding respective task
24 selections.

1 16. The computer program product for managing a project of claim 15, further
2 comprising:

3 an interactive interface definition function, said interactive interface definition
4 function interactively receiving and storing a plurality of said project tracking interface
5 definitions, each project tracking interface definition being associated with a respective
6 group of users of said plurality of users.

1 17. The method of claim 1, wherein each said project tracking interface definition
2 includes a chronological ordering relationship among task selections of its respective set of
3 task selections and at least one indicator indicating a next expected task selection in said
4 chronological ordering relationship among task selections.

1 18. The program product of claim 8, wherein each said project tracking interface
2 definition includes a chronological ordering relationship among task selections of its
3 respective set of task selections and at least one indicator indicating a next expected task
4 selection in said chronological ordering relationship among task selections.